

Kolmogorov-Loveland Sets and Advice Complexity Classes

Thomas Hugel

LIAFA - Université Paris 7 & CNRS - case 7014
F-75205 Paris Cedex 13

Abstract

Loveland complexity Loveland (1969) is a variant of Kolmogorov complexity, where it is asked to output separately the bits of the desired string, instead of the string itself. Similarly to the resource-bounded Kolmogorov sets we define Loveland sets.

We highlight a structural connection between resource-bounded Loveland sets and some advice complexity classes.

This structural connection enables us to map to advice complexity classes some properties of Kolmogorov sets first noticed by Hartmanis Hartmanis (1983) and thoroughly investigated in Longpré's thesis Longpré (1986):

1. Non-inclusion properties of Loveland sets result in hierarchy properties on the corresponding advice complexity classes;
2. Immunity properties of Loveland sets result in the non-existence of natural proofs between the corresponding advice complexity classes, in the sense of Razborov & Rudich Razborov and Rudich (1997).

1 Introduction

Kolmogorov complexity is a measure of algorithmic randomness. If we consider words of length n over the boolean alphabet $\Sigma = \{0, 1\}$, then $KS(x)$ is defined to be the length of the smallest input enabling a given universal Turing machine U to output x . It is possible to “hide” x in the input, so $KS(x) \leq l(x) + O(1)$ (where $l(x)$ is the length of x). Since there are $2^n - 1$ words in $\Sigma^{\leq n-1}$ but 2^n words in Σ^n , by the pigeonhole principle, at least one word of length n cannot be output with a smaller input. Such a word is called *incompressible*. Numerous applications of this result, known as *the Incompressibility Method* are given in the textbook of Li & Vitányi Li and Vitányi (2008). The same argument shows that almost all words are almost incompressible.

The Shannon-Lupanov theorem Shannon (1949); Lupanov (1958) states that any boolean function on n variables can be computed by a circuit of size at most $\frac{2^n}{n} (1 + o(1))$. This bound is tight; moreover almost all boolean functions

have almost this complexity. Trakhtenbrot (1984) drew a parallel between boolean functions having a circuit complexity of at least $(1 - \varepsilon) \frac{2^n}{n}$ and words such that $KS(x) \geq (1 - \varepsilon) l(x)$. Such an analogy suggests that there may be a connection between hard boolean functions and incompressible words.

Indeed, Karp & Lipton noticed that languages in P/\log could be computed by “small circuits with easy descriptions”. Looking for some kind of reciprocal, Hermo & Mayordomo (1994) gave a nice characterization of the advice complexity class P/\log in terms of the resource-bounded Kolmogorov complexity of its circuits. Assuming that $t(n)$ and $s(n)$ are constructible functions, a sequence of words (x_n) of length n belongs to the *Kolmogorov set* $KS[f(n), t(n), s(n) | y(n)]$ iff for each integer n , x_n is computable by the universal Turing machine U with an input of length at most $f(n)$ in time $t(n)$, in space $s(n)$ and with knowledge of some word $y(n)$. Using the P -completeness of the Circuit Value Problem, they proved that P/\log is the set of languages decidable by circuits belonging to $KS[O(\log n), \text{poly}(n), +\infty | n]$ or, equivalently, by circuits belonging to $KS[O(\log n), +\infty, O(\log n) | n]$.

Building on Hartmanis (1983), Longpré (1986) stated many structural properties of the above Kolmogorov sets. However, the characterization of Hermo & Mayordomo does not make it possible to map the properties of the Kolmogorov sets to the advice complexity classes, since several different circuits may compute the same function and not all words encode circuits. To do so one needs a more narrow approach.

The purpose of the present work is to provide a characterization of some advice complexity classes by Kolmogorov sets enabling to map some of the properties of the latter to the former:

1. Non-inclusion properties of Kolmogorov sets result in hierarchies properties on the corresponding advice complexity classes;
2. Immunity properties of Kolmogorov sets result in the non-existence of natural proofs between the corresponding advice complexity classes, in the sense of Razborov & Rudich (1997).

In a breakthrough paper Razborov and Rudich (1997), Razborov & Rudich identified some properties shared by all known proofs of lower bounds on the non-monotone circuit complexity of some individual boolean functions, which they called *natural properties*, and showed that such properties would not be sufficient to derive superpolynomial lower bounds for circuits (with some hardness assumption).

We believe that Kolmogorov sets are a convenient substitute for diagonalization, since they are more compact and explicit. In particular, the results presented in this paper may be viewed as resulting from diagonalization.

2 Connection between Kolmogorov Sets and Advice Complexity Classes

2.1 Framework

Kolmogorov-Loveland Complexity. To make the connection as clear as possible we do not use simple Kolmogorov complexity KS but a variant introduced by Loveland (1969) called *decision Kolmogorov complexity* (KD) in the classification of Uspensky & Shen (1996), and also referred to as *uniform complexity* in Li and Vitányi (1997). In this variant, the universal Turing machine U is not required to output the word x but to give $x[i]$ (the i^{th} bit of x) upon request. The input of U consists thus in a self-delimited tuple $\langle p, i, y \rangle$ where p is the actual input, i is the number of the asked bit of x , and y is a word given for free as an auxiliary input. We take as usual $\langle a, b \rangle = 1^{l(a)}0ab$ and $\langle a, b, c \rangle = \langle \langle a, b \rangle, c \rangle$.

Kolmogorov-Loveland Sets. In the setting of Kolmogorov-Loveland complexity, the required output is a single bit, so it does not make sense to measure time and space with respect to the output. Instead we are going to measure them with respect to the input. This will allow a simpler connection with advice complexity classes. So we say that a sequence of words x_n of length n belongs to $KD[f(n), t(u), s(u) | y(n)]$ iff for each integer n there exists a program $p(n)$ of length at most $f(n)$ such that for all i between 0 and $n-1$, $U(\langle p(n), i, y(n) \rangle) = x_n[i]$ and the computation is done in time at most $t(u)$ and in space at most $s(u)$, where $u = l(\langle p(n), y(n) \rangle)$. The choice of u may seem rather strange at first glance, but we choose this definition so that the connection with advice complexity classes may be as simple as possible. Moreover bounding resources with respect to input size makes this complexity *prefix-monotonic*, in the sense that the program p and the word y used for a word x in time $t(u)$ and in space $s(u)$ can also be used for all of x 's prefixes.

Advice Complexity Classes. The advice complexity classes we are going to consider are of the form $DTISP(t(u), s(u)) / f(n)$. Given two fully time and space constructible functions $t(u)$ and $s(u)$, a language L over the alphabet $\Sigma = \{0, 1\}$ is in $DTISP(t(u), s(u))$ iff there exists a Turing machine deciding it in time $O(t(u))$ and in space $O(s(u))$. Now the advice complexity classes are defined as follows: let \mathcal{C} be a complexity class. $L \in \mathcal{C} / f(n)$ iff there exists a language L' in \mathcal{C} such that for all integer n , there exists a word w_n (the advice) of length at most $f(n)$ such that for any word x of length n , $x \in L \iff \langle w_n, x \rangle \in L'$.

Characteristic Words Sequence of a Language. Given a language L , we consider its *characteristic words sequence*, defined as follows. First we recall that a word x of length n over the alphabet Σ can be considered (in the binary numeral system) as an integer $\text{int}(x)$ between 0 and $2^n - 1$. We denote the

inverse function from integers to words as $\text{word}_n(i)$. Note that $\text{word}_n(i)$ will have length n and may begin by a sequence of zeros. In particular the word 0^n corresponds to the integer 0 and the word 1^n to the integer $2^n - 1$ (keep this in mind). Now the characteristic word of the language L for length n is a word of length $N = 2^n$ defined by $L_n[i] = \mathbf{1}_{\text{word}_n(i) \in L}$ (i.e. 1 if $\text{word}_n(i) \in L$ and 0 otherwise).

2.2 Connection Lemma

Lemma 1. *Let $f(n)$, $t(u)$ and $s(u)$ be integer functions such that $t(u)$ and $s(u)$ are non-decreasing and fully time and space constructible, $t(u) \geq u$ and $s(u) \geq \log u$:*

1. *if $L \in DTISP(t(u), s(u)) / f(n)$, then $(L_n) \in KD[f(\log N) + O(1), O(t(u) \log t(u)), O(s(u)) | 1^{\log N}]$*
2. *if $(L_n) \in KD[f(\log N), t(u), s(u) | 1^{\log N}]$, then $L \in DTISP(t(u), s(u)) / f(n)$.*

Proof. The main idea is to switch from integers to strings of a given length and vice-versa.

1. if $L \in DTISP(t(u), s(u)) / f(n)$, then there exists a language L' decided by a Turing machine M' in time $O(t(u))$ and in space $O(s(u))$ such that for all integer n , there exists an advice word w_n of length at most $f(n)$ such that $x \in L \iff \langle w_n, x \rangle \in L'$. Now consider the Turing machine M which on input $\langle p, i, y \rangle$ computes $n = l(y)$ and $x = 0^{n-l(i)}i$, and simulates M' on $\langle p, x \rangle$. By construction, $M \langle w_n, i, y \rangle = L_n[i]$ and M runs in time $O(u + t(u)) = O(t(u))$ and in space $O(\log u + s(u)) = O(s(u))$. Now M can be simulated by the universal Turing machine U (which has a fixed number of tapes) in time $O(t(u) \log t(u))$ and in space $O(s(u))$ thanks to the simulation method of Hennie & Stearns Hennie and Stearns (1966).
2. if $(L_n) \in KD[f(\log N), t(u), s(u) | 1^{\log N}]$, then for all integer n , there exists a program p_n of length at most $f(n)$ such that for all integer i between 1 and N , $U \langle p_n, i, 1^n \rangle = L_n[i]$. Now p_n can be used as an advice, as follows: consider the Turing machine M which on input $\langle w, x \rangle$ computes $n = l(x)$, $i = x$ without the initial zeros, and simulates U on $\langle w, i, 1^n \rangle$. Then $M \langle p_n, x \rangle = \mathbf{1}_{x \in L}$ and this computation is done in time $O(u + t(u)) = O(t(u))$ and in space $O(\log u + s(u)) = O(s(u))$.

□

3 Transfer of Properties

We are going to use the above Connection Lemma to transfer properties of Loveland sets to the corresponding advice complexity classes:

1. Non-inclusion properties of Loveland sets result in hierarchy properties on the corresponding advice complexity classes;
2. Immunity properties of Loveland sets result in the non-existence of natural proofs between the corresponding advice complexity classes.

So we shall first establish the properties of Loveland sets. They are very similar to the properties established by Longpré in his thesis, but we must revisit them because we work with Loveland complexity instead of standard Kolmogorov complexity, and moreover we measure the resources with respect to the input instead of the output.

3.1 Non-Inclusions and Hierarchies

3.1.1 Sensitivity to Advice Length.

The following proposition is analogous to corollary 3.2 of Longpré's thesis Longpré (1986). Here the proof is simpler because KD is prefix-monotonic.

Proposition 2. *If $f(n) < n$, then there exists a constant c such that for all $y(n)$ we have*

$$KD[f(n) + c, cu \log u, c \log u | y(n)] \not\subseteq KD[f(n), +\infty, +\infty | y(n)] .$$

Proof. By counting, there exists some x of length $f(n) + 1$ incompressible with respect to $y(n)$; by prefix-monotonicity, $z = x0^{n-f(n)-1} \notin KD[f(n), +\infty, +\infty | y(n)]$. Now the program which on input $\langle x, i, y \rangle$ prints $x[i]$ if $i < l(x)$ and 0 otherwise, has length $f(n) + O(1)$, works in time $O(l(x) + l(y))$ and in space $O(\log l(x))$. As above, this program can be simulated by U in time $O(u \log u)$ and in space $O(\log u)$, where $u = l(\langle x, y \rangle)$. So there exists a constant c such that $z \in KD[f(n) + c, cu \log u, c \log u | y(n)]$. \square

Now using this proposition together with our Connection Lemma yields the following result, which was already present in Hermo and Mayordomo (1994) in a similar form. We denote by REC the class of recursive languages.

Theorem 3. *Let $f(n)$ and $g(n)$ be two integer functions such that $f(n) + \frac{g}{2}(n) < 2^n$ and $g(n)$ is non-decreasing and unbounded. Then*

$$DTISP(u \log u, \log u) / (f(n) + g(n)) \not\subseteq REC / f(n) .$$

Proof. By proposition 2 and the fact that $g(n)$ is non-decreasing and unbounded, we consider a sequence of words L_n of length $N = 2^n$ belonging to $KD[(f + g)(\log N), cu \log u, c \log u | 1^{\log N}] \setminus KD[(f + \frac{g}{2})(\log N), +\infty, +\infty | 1^{\log N}]$ for some c and for n large enough. Now by the above Connection Lemma, the corresponding language L is in $DTISP(u \log u, \log u) / (f(n) + g(n))$. Suppose by contradiction that $L \in REC / f(n)$. By the Connection Lemma, there exists a constant c' such that (L_n) is in $KD[f(\log N) + c', +\infty, +\infty | 1^{\log N}]$. For n large enough, $g(n) > 2c'$ and then (L_n) is in $KD[(f + \frac{g}{2})(\log N), +\infty, +\infty | 1^{\log N}]$, a contradiction. \square

3.1.2 Sensitivity to Time and Space.

We give a proposition analogous to theorems 4.3 and 4.4 of Longpré's thesis Longpré (1986). The main difference here is that we consider that resources are bounded with respect to the input rather than the output.

Proposition 4. *Let $f(n)$, $t(u)$ and $s(u)$ be integer non-decreasing and constructible functions such that $f(n) < n$, $t(u) \geq u$ and $s(u) \geq \log u$. Then there exists some constant c such that if $t'(u) \geq c 2^{f(2^u)} f(2^u) t(2f(2^u) + u) (f(2^u) + \log t(2f(2^u) + u))$ and $s'(u) \geq c (2^{f(2^u)} f(2^u) + s(2f(2^u) + u))$, for n large enough,*

$$KD[c, t'(u), s'(u) | n-1] \not\subseteq KD[f(n), t(u), s(u) | n-1] .$$

Proof. By counting, there must exist some x of length $f(n) + 1$ incompressible with respect to $n - 1$ in time $t(u)$ and in space $s(u)$; by prefix-monotonicity, $z = x0^{n-f(n)-1} \notin KD[f(n), t(u), s(u) | n-1]$. Finding the smallest such x in the lexicographic order can be performed by exhaustive search by running all programs of length at most $f(n)$ in time $t(u)$ and in space $s(u)$. Here we face a trade-off between time and space: if we choose to store all generated strings to avoid recomputations, this increases the required space; otherwise we may iterate the exhaustive search for each string of length $f(n) + 1$ until we find the desired one. The first option takes:

- an overall advice of length $O(1)$ (since $n - 1$ is given for free);
- an overall time of $O(2^{f(n)} f(n) t(2f(n) + \log n + 2))$, since $l(\langle a, b \rangle) = 2l(a) + l(b) + 1$ and $l(n - 1) \leq \log n + 1$; now $u = l(\langle p, n - 1 \rangle) > \log n + 1$. So the time bound is $O(2^{f(2^u)} f(2^u) t(2f(2^u) + u))$. Again there is an extra logarithmic factor due to the simulation by our fixed machine U ;
- an overall space of $O(2^{f(n)} f(n) + s(2f(n) + \log n + 2))$, i.e. $O(2^{f(2^u)} f(2^u) + s(2f(2^u) + u))$.

□

Now using this proposition together with our Connection Lemma yields the following result.

Theorem 5. *Let $f(n)$, $g(n)$, $t(u)$ and $s(u)$ be integer non-decreasing and constructible functions such that $f(n) + g(n) < 2^n$, g is unbounded, $t(u) \geq u$ and $s(u) \geq \log u$. Let $t''(u)$ and $s''(u)$ be such that $t''(u) = \omega(t(u) \log t(u))$ and $s''(u) = \omega(s(u))$. Then there exists some constant c such that if $t'(u) \geq c 2^{(f+g)(u)} (f+g)(u) t''(2(f+g)(u) + u) ((f+g)(u) + \log t''(2(f+g)(u) + u))$ and $s'(u) \geq c (2^{(f+g)(u)} (f+g)(u) + s''(2(f+g)(u) + u))$, then*

$$DTISP(t'(u), s'(u)) / c \not\subseteq DTISP(t(u), s(u)) / f(n) .$$

Proof. By proposition 4, we consider a sequence of words L_n of length $N = 2^n$ belonging to $KD[c, t'(u), s'(u) | 1^{\log N}] \setminus KD[(f+g)(\log N), t''(u), s''(u) | 1^{\log N}]$ for some c and for n large enough. Now by the Connection Lemma, the corresponding language L is in $DTISP(t'(u), s'(u)) / c$. Suppose by contradiction that $L \in DTISP(t(u), s(u)) / f(n)$. By the Connection Lemma, there exists a constant c' such that (L_n) is in $KD[f(\log N) + c', c't(u) \log t(u), c's(u) | 1^{\log N}]$. For n large enough, $(f+g)(\log N) \geq f(\log N) + c'$, $t''(u) \geq c't(u) \log t(u)$ and $s''(u) \geq c's(u)$, so (L_n) is in $KD[(f+g)(\log N), t''(u), s''(u) | 1^{\log N}]$, a contradiction. \square

3.2 Immunity and Natural Proofs

3.2.1 Immunity of Kolmogorov Sets.

Immunity is an indication that a language is algorithmically very complex, in the sense that given a complexity class \mathcal{C} , a language L is called \mathcal{C} -immune iff L is infinite and does not have any infinite subset belonging to \mathcal{C} . To this we add the notion of density: a language L has *partial density* δ iff there exist infinitely many n 's such that L contains at least $\delta(n)2^n$ words of length n . Thus we generalize the notion of immunity using density: we say that a language L is \mathcal{C} -immune for partial density δ iff L is infinite and does not have any infinite subset of partial density at least δ belonging to \mathcal{C} .

As noted by Hartmanis (1983) and further developed by Longpré (1986), the complements of Kolmogorov sets are immune. Longpré's results for immunity (theorems 3.7, 3.8 and 4.13 of Longpré (1986)) concern classical complexity classes and global density. Here we deal with advice complexity classes and partial density, as follows.

Proposition 6. *Let $f(n)$, $g(n)$, $t(u)$ and $s(u)$ be integer non-decreasing and constructible functions, such that $f(n) < n$, $g(n)$ is unbounded and $g(n) < f(n)$. Let $\delta(n)$ be a function to the real interval $[0, 1]$ and $\rho(n) = (1 - \delta(n))2^n + 1$. If $t'(u) \geq u$ and $s'(u) \geq u$ are non-decreasing, $t'(u) = o\left(\frac{t(\log u)}{\rho(u) \log(\rho(u)t(\log u))}\right)$ and $s'(u) = o(s(\log u))$, then $\Sigma^* \setminus \bigcup_{n \in \mathbb{N}} KD[f(n), t(u), s(u) | n-1]$ is $DTISP(t'(u), s'(u)) / (f-g)(n)$ -immune for partial density δ .*

Proof. Let us consider any infinite language $A \in DTISP(t'(u), s'(u)) / (f-g)(n)$ with partial density δ . We argue that for n large enough, the lexicographically smallest word of length n belonging to A is in $KD[f(n), t(u), s(u) | n-1]$. Indeed there exists a Turing machine M working in time $t'(u)$ and in space $s'(u)$, and a sequence of advice (w_n) of length $l(w_n) \leq (f-g)(n)$ such that for all $x \in \Sigma^n$, $x \in A \iff M\langle w_n, x \rangle = 1$. Thus it suffices to simulate M with advice w_n on all x 's of length n in the lexicographic order.

- This can be done by a program of length $(f-g)(n) + O(1)$ (since $n-1$ is given for free).

- For an n such that A contains at least $\delta(n) 2^n$ words of length n , there are at most $(1 - \delta(n)) 2^n + 1 = \rho(n)$ steps of simulation, each step requiring a time $t'(v)$ where $v = l(\langle w_n, 1^n \rangle) \leq 2l(w_n) + n + 1 \leq 2^{2l(w_n) + \log(n-1) + 1} = 2^{l(\langle w_n, n-1 \rangle)} = 2^u$. The simulation by our universal Turing machine U can thus be performed in time $O((\rho t' \log(\rho t'))(2^u)) = o\left(\frac{t(u)}{\log(\rho(2^u)t(u))} \log \frac{t(u)}{\log(\rho(2^u)t(u))}\right)$. Since $\frac{t(u)}{\log(\rho(2^u)t(u))} \leq t(u) \leq \rho(2^u)t(u)$, this is $o(t(u))$.
- The above simulation can be performed in space $O(v + s'(v)) = O(s'(2^u)) = o(s(u))$.

□

3.2.2 Non-Existence of Natural Proofs among Advice Complexity Classes.

We first recall the definitions of Razborov and Rudich (1997) (section 2.2). A *combinatorial property* is a set of boolean functions. Each of the 2^{2^n} boolean functions on n -bit inputs can be described by a binary word of length 2^n (which in turn can be seen as the characteristic word of a language, see section 2.1 above). Thus a combinatorial property can be seen as a language with words of length powers of 2. The question whether a given boolean function belongs to a combinatorial property is an algorithmic problem which requires some time and space depending on the length 2^n of the boolean functions. Thus it is possible to group together combinatorial properties with respect to this algorithmic complexity, and such sets of combinatorial properties are some kinds of *complexity classes*. These complexity classes should not be confused with the complexity classes of the boolean functions themselves, i.e. the time and space (depending on n) required to compute the boolean functions on n -bit inputs!

Given a complexity class \mathcal{C} , a combinatorial property Γ is called *\mathcal{C} -natural* for *partial density* δ iff there exists $\Xi \subseteq \Gamma$ such that:

constructibility: $\Xi \in \mathcal{C}$

largeness: Ξ is infinite and has partial density $\delta \circ \log$ (since the words in Ξ have lengths of the form 2^n)

In fact the density considered in Razborov and Rudich (1997) is global, but we refine it to use partial density. So, what does a non-natural property look like? It is a property without any large constructible sub-property. This looks very much like the aforementioned notion of immunity. Indeed:

Remark 7. Let Γ be a combinatorial property, \mathcal{C} a complexity class and $\delta : \mathbf{N} \rightarrow [0, 1]$. Then Γ is not \mathcal{C} -natural for partial density δ iff Γ is \mathcal{C} -immune for partial density $\delta \circ \log$.

Now what is the use of a combinatorial property? Given a complexity class \mathcal{D} , an infinite combinatorial property Γ is called *useful* against \mathcal{D} iff

usefulness: given a sequence (L_n) of characteristic words, if $L_n \in \Gamma$ infinitely often then $L \notin \mathcal{D}$.

Why is it called “usefulness”? Because in order to prove that $L \notin \mathcal{D}$, it is enough to prove that $L_n \in \Gamma$ infinitely often. So Razborov & Rudich manage to prove that for various circuit complexity classes \mathcal{C} and \mathcal{D} there are no \mathcal{C} -natural properties against \mathcal{D} . Using proposition 6 together with our Connection lemma, we prove the following result:

Theorem 8. *Let $f(n)$, $g(n)$, $t(u)$ and $s(u)$ be integer non-decreasing and constructible functions, such that $f(n) < 2^n$, g is unbounded and $g(n) < f(n)$. Let $\delta(n)$ be a function to the real interval $[0, 1]$ and $\rho(n) = (1 - \delta(\log n)) 2^n + 1$. If $t'(u) \geq u$ and $s'(u) \geq u$ are non-decreasing, $t'(u) = o\left(\frac{t(\log u)}{\rho(u) \log(\rho(u)t(\log u))}\right)$ and $s'(u) = o(s(\log u))$, then*

there is no $DTISP(t'(u), s'(u)) / (f - g)(\log n)$ -natural property
for partial density δ useful against $DTISP(t(u), s(u)) / f(n)$.

Proof. Let Γ be a $DTISP(t'(u), s'(u)) / (f - g)(\log n)$ -natural property for partial density δ . It is important to notice that all words in Γ have lengths of the form $N = 2^n$. There exists $\Xi \subseteq \Gamma$ such that $\Xi \in DTISP(t'(u), s'(u)) / (f - g)(\log N)$ and Ξ has partial density $\delta \circ \log$. Suppose by contradiction that $\Delta = \Xi \cap \left(\bigcup_{N \in \{2^n\}_{n \in \mathbb{N}}} KD[f(\log N), t(u), s(u) | 1^{\log N}] \right)$ is finite. Then $\Xi \setminus \bigcup_{N \in \{2^n\}_{n \in \mathbb{N}}} KD[f(\log N), t(u), s(u) | 1^{\log N}]$ is still infinite with partial density $\delta \circ \log$ and is still in $DTISP(t'(u), s'(u)) / (f - g)(\log N)$, which contradicts proposition 6. Thus Δ is infinite, and there are infinitely many n 's such that we can pick in this set an element L_n of length $N = 2^n$. Now by the Connection Lemma, the language L having these L_n 's as characteristic words is in $DTISP(t(u), s(u)) / f(n)$. Thus Γ is not useful against $DTISP(t(u), s(u)) / f(n)$. \square

4 Conclusion

We had to take resource bounds on Kolmogorov-Loveland complexity with respect to the input in order to make a straightforward connection with the advice complexity classes. This led to use both n and u in the complexity classes of the form $DTISP(t(u), s(u)) / f(n)$, and we had to make frequent and inelegant conversions between n and u . To make things clearer, we suggest that in the advice complexity classes, resource bounds on $\langle w, x \rangle$ should be taken with respect to x only. This would not change the main classes P/poly and P/\log .

One may object that our results relativize. They do as do all diagonalization results, since our results are some kinds of diagonalizations. That is also the reason why we think that it would be surprising if one could do better than exhaustive search in our simulations.

However we believe that it is important to exhibit separation results, even if they are simple. Indeed we want to recall for example that in the deep proof that $DLIN \neq NLIN$ of Paul et al. (1983), the only separation result invoked is a simple time hierarchy on alternating Turing machines obtained by diagonalization.

Acknowledgments

This research was done while the author was an intern student at LRI (Orsay, France) under supervision of Sophie Laplante. The author is very grateful to Sophie Laplante for many helpful discussions.

References

- Hartmanis, J., Nov. 1983. Generalized Kolmogorov complexity and the structure of feasible computations. In: 24th Annual Symposium on Foundations of Computer Science (FOCS 1983). IEEE, pp. 439–445.
URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4568108>
- Hennie, F. C., Stearns, R. E., 1966. Two-tape simulation of multitape Turing machines. *J. ACM* 13 (4), 533–546.
- Hermo, M., Mayordomo, E., Jul. 1994. A Note on polynomial-size circuits with low resource-bounded Kolmogorov complexity. *Mathematical Systems Theory* 27 (4), 347–356.
URL <http://www.springerlink.com/index/10.1007/BF01192144>
- Li, M., Vitányi, P., 1997. *An Introduction to Kolmogorov Complexity and Its Applications*, 2nd Edition. Springer-Verlag, New-York.
URL citeseer.ifi.unizh.ch/li97introduction.html
- Li, M., Vitányi, P., 2008. *An Introduction to Kolmogorov Complexity and its Applications*, 3rd Edition. Springer Verlag.
- Longpré, L., 1986. *Resource Bounded Kolmogorov Complexity, A Link between Computational Complexity and Information Theory*. Ph.D. thesis, Cornell University.
- Loveland, D. W., Dec. 1969. A Variant of the Kolmogorov Concept of Complexity. *Information and Control* 15 (6), 510–526.
URL <http://linkinghub.elsevier.com/retrieve/pii/S0019995869905385>
- Lupanov, O. B., 1958. A Method for Synthesizing Circuits. *Izv. vysshikh uchebnykh zavedenii, Radiofizika* 1, 120–140.
- Paul, W. J., Pippenger, N., Szemerédi, E., Trotter, W. T., Nov. 1983. On determinism versus non-determinism and related problems. In: 24th Annual Symposium on Foundations of Computer Science (FOCS 1983). IEEE, pp. 429–438.
URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4568107>
- Razborov, A. A., Rudich, S., Aug. 1997. Natural Proofs. *Journal of Computer and System Sciences* 55 (1), 24–35.
- Shannon, C. E., 1949. The synthesis of two-terminal switching circuits. *Bell System Technical Journal* 28, 59–98.

Trakhtenbrot, B. A., Oct. 1984. A Survey of Russian Approaches to Perebor (Brute-Force Searches) Algorithms. IEEE Annals of the History of Computing 6 (4), 384–400.

URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4640789>

Uspensky, V. A., Shen, A., Jun. 1996. Relations between varieties of Kolmogorov complexities. Mathematical Systems Theory 29 (3), 271–292.

URL <http://www.springerlink.com/index/10.1007/BF01201280>